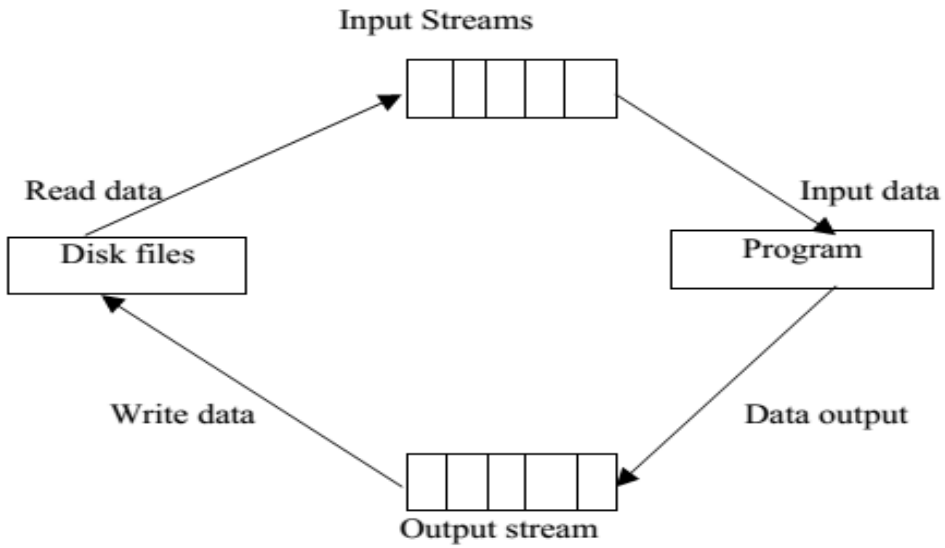
Unit-8 File and Streams

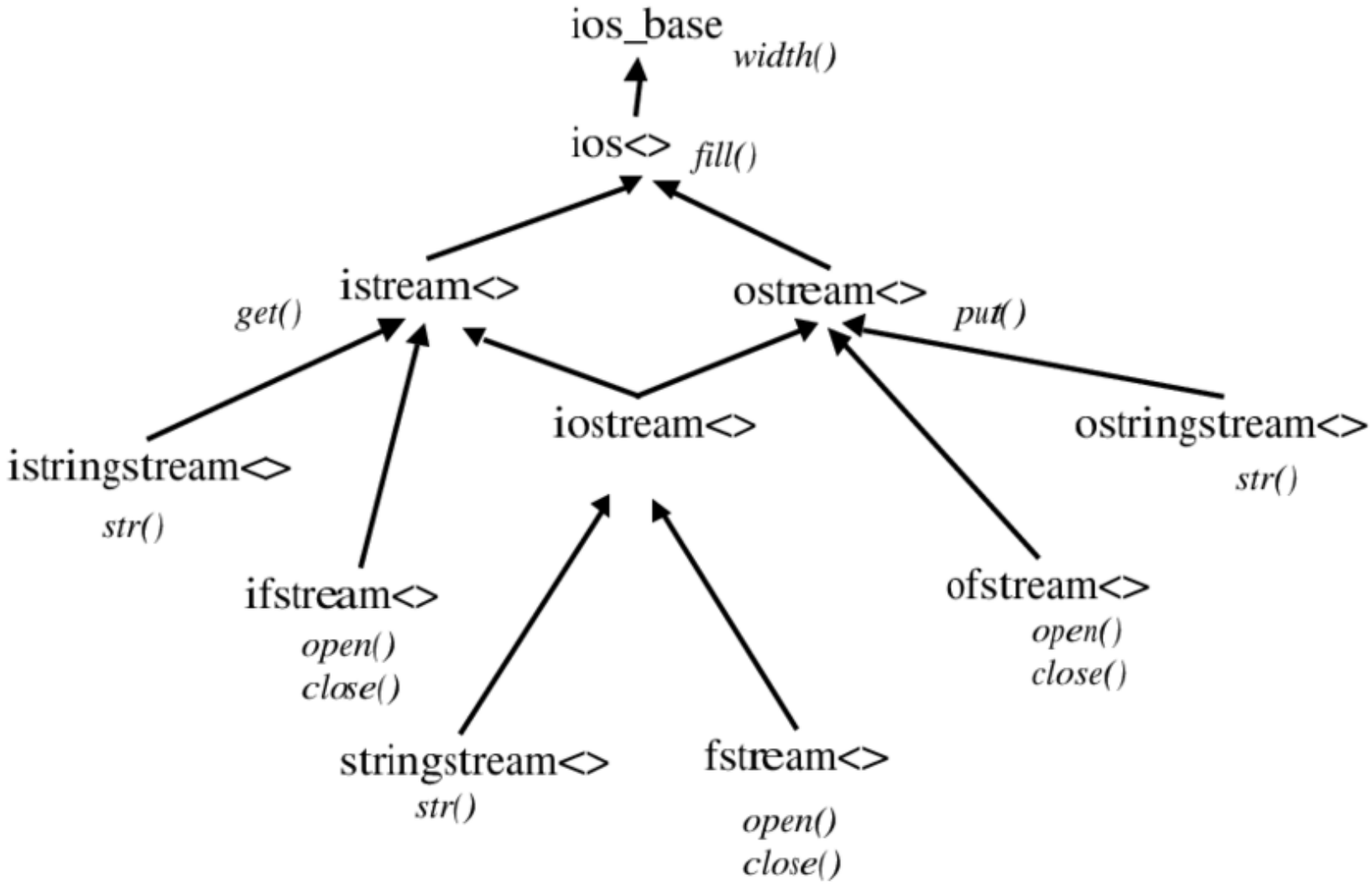
Introduction:

- Three data types for writing program in files:
  - fstream : supports for simultaneously input and output operations on files.
  - ifstream: it provides input operation on files.
  - ofstream: provides output operation on files.



File Input and output stream

Stream classes:



Opening a File:

- A file must be opened before you can read from it or write to it.
- ofstream or fstream object may be used to open a file for write.
- ifstream object is used to open a file for reading purposes only.
- Syntax: for open function

void open(const char *file_name, ios::open_mode mode);

Where “const char *file_name” specifies the name & location of file to be opened.

“ios::open_mode mode” defines mode in which file should be opened.

File Mode Flags

Mode Flag	Meaning
ios::app	Append mode. If the file already exists, its contents are preserved and all output is written to the end of the file.
ios::ate	If the file already exists, the program goes directly to the end of it.
ios::binary	Binary mode. Information is written to or read from the file in pure binary format.
ios::in	Input mode. Information is read from the file.
ios::nocreate	If the file does not already exist, the open function call fails.
ios::noreplace	If the file exists, the open function call fails.
ios::out	Output mode. Information is written to the file.
ios::trunc	If the file exists, its contents are deleted.

Eg:

```
ofstream outfile;
outfile.open(“file.data”, ios::out/ios::trunc);

fstream afile;
afile.open(“file.data”, ios::out/ios::in);
```

Closing the File:

- automatically closes all the opened files and releases all the allocated file.
- Syntax: void close();

Writing to a file:

- We use stream insertion operator(<<) the only difference is we use ofstream or fstream object instead of cout object.
- Eg: ofstream<<“a”;

Reading from a file:

- we use stream extraction operator(>>) here we use ifstream or fstream object instead of cin object.

Eg: fstream>>a

i/o file:

```cpp
#include <fstream>
#include <iostream>

using namespace std;

int main()
{
  char str[10];
  ofstream a_file ( "example.txt" ); //Creates an instance of ofstream, and opens example.txt
  a_file<<"This text will now be inside of example.txt";// Outputs to example.txt through a_file
  a_file.close(); // Close the file stream explicitly
  ifstream b_file ( "example.txt" ); //Opens for reading the file
  b_file>> str; //Reads one string from the file
  cout<< str <<"\n";//Should output 'this'
  cin.get();    // wait for a keypress
  // b_file is closed implicitly here
}
```

input

```
This
```

Reading and Writing text file

```cpp
#include <iostream>
#include <fstream>
using namespace std;

int main() {
  fstream ob;

  ob.open("test.txt", ios::out);  // opening file in writing mode

  ob << "hello world\n";  // writing data to file

  ob << "this is my first file";

  ob.close();  // closing the file

  ob.open("test.txt", ios::in);  // again opening the file but in reading mode

  while (!ob.eof()) {
    string str;

    ob >> str;  // reading word by word from file and storing in str

    cout << str << "\n";  // printing str
  }

  ob.close();  // closing the file after use

  return 0;
}
```

input

```
hello
world
this
is
my
first
file
```

Reading and Writing Binary file

```
#include <iostream>
#include <fstream>
#define FILE_NAME "emp.dat"
using namespace std;
class Employee {
private :
    int     empID;
    char    empName[100] ;
    char    designation[100];
    int     ddj,mmj,yyj;
    int     ddb,mmb,yyb;
public  :
    void readEmployee(){
        cout<<"EMPLOYEE DETAILS"<<endl;
        cout<<"ENTER EMPLOYEE ID : " ;
        cin>>empID;
        cin.ignore(1);
        cout<<"ENTER  NAME OF THE EMPLOYEE : ";
        cin.getline(empName,100);
        cout<<"ENTER DESIGNATION : ";
        cin.getline(designation,100);
        cout<<"ENTER DATE OF JOIN:"<<endl;
        cout<<"DATE : "; cin>>ddj;
        cout<<"MONTH: "; cin>>mmj;
        cout<<"YEAR : "; cin>>yyj;
        cout<<"ENTER DATE OF BIRTH:"<<endl;
        cout<<"DATE : "; cin>>ddb;
        cout<<"MONTH: "; cin>>mmb;
        cout<<"YEAR : "; cin>>yyb;  }
    void displayEmployee(){
        cout<<"EMPLOYEE ID: "<<empID<<endl
         <<"EMPLOYEE NAME: "<<empName<<endl
         <<"DESIGNATION: "<<designation<<endl
         <<"DATE OF JOIN: "<<ddj<<"/"<<mmj<<"/"<<yyj<<endl
         <<"DATE OF BIRTH: "<<ddb<<"/"<<mmb<<"/"<<yyb<<endl;    }};
```

```
int main(){
    Employee emp;
    emp.readEmployee();
    fstream file;
    file.open(FILE_NAME,ios::out|ios::binary);
    if(!file){
        cout<<"Error in creating file...\n";
        return -1;}
    file.write((char*)&emp,sizeof(emp));
    file.close();
    cout<<"Date saved into file the file.\n";
    file.open(FILE_NAME,ios::in|ios::binary);
    if(!file){
        cout<<"Error in opening file...\n";
        return -1;}
    if(file.read((char*)&emp,sizeof(emp))){
            cout<<endl<<endl;
            cout<<"Data extracted from file..\n";
            //print the object
            emp.displayEmployee();}
    else{
        cout<<"Error in reading data from file...\n";
        return -1;
    file.close();
    return 0;
}
```

Random Access File

- Instant Access
  - Want to locate records quickly?
    - Airline reservation, Banking System, ATMs.
  - Sequential files must search through each one
- Random access files are the solution
    - Instant Access
    - Update/delete items without changing other data.
- C++ imposes no structure on files.
    - The programmer must create random access files.
    - Fixed length records.

Fill in the Blanks

1. The functions `open()` and `close()` are defined in _____ class.
2. Opening a file in `ios::out` mode also opens a file in _____ mode.
3. The statement `ifile.seekg(0,ios::cur)` sets the get pointer at _____ position.
4. Command-line arguments are accessed through arguments to _____ .
5. Mode bits such as `app` and `ate` are defined in _____ class.

Multiple Choice Questions

1. The function `fail()` returns a non-zero value if
   (a) failbit is set (b) badbit is set (c) hardbit is set (d) failbit or badbit is set
2. Which of these should be used to write data that contains variables of type `float` to a file?
   (a) `seekp()` (b) `write()` (c) `put()` (d) insertion operator
3. Which of these headers is used for disk I/O operations?
   (a) `iostream` (b) `fstream` (c) both (a) and (b) (d) none
4. The data can be outputted to an object of the class `ofstream` using the insertion operator (`<<`) because
   (a) the insertion operator works with all classes
   (b) the `ofstream` class is a stream
   (c) the insertion operator is overloaded in `ofstream`
   (d) the output is actually sent to console
5. The statement `ofile.write((char*)&obj, sizeof(obj))` writes
   (a) the data in `obj` to `ofile`
   (b) the member functions of `obj` to `ofile`
   (c) the address of `obj` to `ofile`
   (d) the data and the member functions of `obj` to `ofile`

1 fstream 2 ios::trunc 3 current 4 main() 5 ios

1 d, 2 b, 3 b, 4 c, 5 a